أولاً: المصفوفات ذات البعد الواحد

وهي مجموعة من الخزانات الذاكرة المتتالية وراء بعضها البعض التي لها نفس الاسم ونفس النمط وتختلف عن بعضها البعض بدليل ويتم التصريح عن المصفوفات ذات البعد الواحد بالشكل:

نمط المعطيات اسم المصفوفة [عدد العناصر];

مثال:

```
int a[4]
```

هنا تم التصريح عن مصفوفة اسمها a مكونة من أربعة عناصر كل عنصر من عناصرها عدد صحيح ويتم التعبير عن عناصر المصفوفة بالشكل:

```
a[0]   a[1]   a[2]   a[3]
```

* ملاحظات:

1- يمكننا التصريح عن المصفوفة وإعطاؤها قيم بنفس الوقت

```
int a[4] = {2, -5, 4, 1}
```

2- اذا كانت عدد القيم المعطاة أقل من عدد عناصر المصفوفة فإنه يتم اعطاء بقية العناصر القيمة 0:

```
int a[4] = {2, 1}
```

3- اذا تم اعطاء قيم أكبر من عدد عناصر المصفوفة فإننا نحصل على خطأ.

```
int a[4] = {1, 2, 9, 8, 3, 2}
```

4- اذا أردنا اعطاء جميع عناصر المصفوفة القيمة صفر نكتب

```
int a[4] = [0];
```

5- اذا لم يتم تحديد بعد المصفوفة فيتم أخذ البعد حسب القيم المعطاة.

```
int a[ ] = {7, 3, 2, 9}
```

6- لادخال أو طباعة عناصر مصفوفة ذات بعد واحد نستخدم حلقة for أو أي حلقة تكرارية

اكتب برنامج يقوم بإيجاد اصغر قيمة من عناصر مصفوفة ذات بعد واحد.

```
#include<iostream.h>
main()
{ const int n=5;
  int a[n];
  int min;
  for(int i=0;i<n;i++)
  { cout<<"a["<<i<<"]="; cin>>a[i]; }
  min = a[0];
  for(int i=1;i<n;i++)
    if (a[i] < min)
```

```
min = a[i];
Cout << "min = " << min;
return 0;
}
```

اكتب برنامج يقوم بإيجاد حاصل ضرب مصفوفتين ذات بعد واحد

```
#include <iostream.h>
main()
{ const int n = 6;
  int a[n];
  int b[n];
  int c;
  for(int i=0; i<n; i++)
  { Cout<<"a["<<i <<"]="; Cin>>a[i]; }
  for(int i=0; i<n; i++)
  { Cout <<"b["<<i<<"]=";
     Cin >>b[i];
  }
  c=0;
  for(int i=0; i<n; i++)
    C= C + a[i] * b[i];
    Cout<< "C="<< C;
    return 0;
}
```

اكتب برنامج يقوم بإيجاد المتوسط الحسابي للعناصر بمصفوفة ذات الدليل الزوجي

```
#include <iostream.h>
main()
{ const int n=6;
  int sum;
  int r=0;
  int a[n]; float avg;
  for(int i=0; i<n; i++)
  { Cout <<"a["<<i<<"]=";
    Cin>>a[i];
  }
```

```
sum = 0;
for(int i=0; i<n; i++)
  if( i%2 == 0)
  { sum = sum + a[i]
      r = r+1;
  }
  average = (float) sum/r;
  Cout << "Avg = " << avg;
 return 0;
}
```

اكتب برنامج يقوم بالمبادلة بين العنصر الأول والأخير، والثاني وما قبل الأخير، وهكذا...

```
#include <iostream.h>
main()
{ const int n=6;
  int a[n]; int x;
  for(int i=0; i<n; i++)
  { Cout << "a[" << i << "]="; Cin >> a[i];
  }
 for(int i=0; i<n/2; i++)
 { x = a[i];
     a[i] = a[(n-1)-i];
     a[(n-1)-i] = x;
  }
  for(int i=0; i<n; i++)
  { Cout << "a[" << i << "]=" << a[i];
    Cout << "\n"; }
return 0;
}
```

* المصفوفات ذات البعدين :

يتم التصريح عن المصفوفة ذات البعدين بالشكل :

نوع المعطيات اسم المصفوفة [عدد الأسطر] [عدد الأعمدة] ;

(عدد الأسطر ← بعد أول ، عدد الأعمدة ← بعد ثاني)

مثال : `int a[2][3]`

هنا تم التصريح عن مصفوفة اسمها a مكونة من سطرين و ثلاثة أعمدة ويتم التعبير عن عناصرها بالشكل :

a[0][0]	a[0][1]	a[0][2]
a[1][0]	a[1][1]	a[1][2]

* ملاحظات :

1- لإدخال وطباعة عناصر مصفوفة ذات بعدين نحتاج إلى حلقتين تكراريتين (حلقتي for)

2- يمكن التصريح عن المصفوفة ذات البعدين وإعطائها قيم بنفس الوقت

```
int a[2][2] = {{7, 1, -9}, {3, 2, 9}, {10, 1, 5}};
```

3- إذا كان عدد القيم أقل من عدد العناصر فبقية العناصر تأخذ القيمة 0

```
int a[2][3] = {{2, 1}, {5, 1}};
```

4- يمكن التصريح عن مصفوفة وإعطائها قيم بالشكل التالي :

```
int [2][3]a = {7, 1, 5, 2, -3, 10};
```

اكتب برنامج للتعامل مع المصفوفات يقوم بما يلي:

1- طباعة عناصر القطر الرئيسي

2- حساب مجموع عناصر القطر الثانوي

3- ضرب عناصر العمود الأول بعدد

4- حساب العاملي لكل عنصر في القطر الرئيسي

5- إيجاد أصغر قيمة من العناصر الواقعة تحت القطر الثانوي

```
#include<iostream.h>
main()
{const int n=3;  int a[n][n];
 int f, s, x, min;
  for(int i=0; i<n; i++)                                    (1)
   for(int j=0; j<n; j++)
    { Cout<<"a["<<i<<"]["<<j<<"]=";
       Cin>>a[i][j];
    }
  for(int i=0; i<n; i++)                                    (2)
    for(int j=0; j<n; j++)
       if(i==j)
        Cout<<"a["<<i<<"]["<<j<<"]="<<a[i][j]<<"\n"
     S=0;                                                   (3)
     for(int i=0; i<n; i++)
      for(int j=0; j<n; j++)
        if(i+j==n-1)    //  شرط القطر الثانوي
          S=S+a[i][j];
         Cout<<"s="<<S<<"\n";
      for(int i=0; i<n; i++)                                (4)
         a[i][0]=x*a[i][0];
      for(int i=0; i<n; i++)      } طباعة العمود الأول
       Cout<<a[i][0]<<"\n";       } الجديد
```

طريقة ثانية:

نوجد الشعاع الناتج عن حاصل ضرب العدد الأول بعدد

```
int c[n];
for(int i=0; i<n; i++)
  c[i] = x * a[i][0];
for(int i=0; i<n; i++)
  cout<<"a["<<i<<"]="<<c[i];

for(int i=0; i<n; i++)
  for(int j=0; j<n; j++)
    if(i<j)          // العناصر فوق القطر الرئيسي
    { f=1;                                          // i<=j
      for(int K=1; K<=a[i][j]; K++)                 // عناصر القطر الرئيسي وما فوقه
        f=f*K;                                      // i>j
        cout<<"f="<<f<<"\n";                        // عناصر تحت القطر الرئيسي
    }                                               // i>=j
                                                    // تحت القطر مع عناصر القطر الرئيسي
min = a[2][1];
for(int i=0; i<n; i++)
  for(int j=0; j<n; j++)
    if(i+j>n-1)     // تحت القطر الثانوي
      if(a[i][j]<min)
        min=a[i][j];
        cout<<"min="<<min;
return 0;
```